Slingshot analysis - Functional Genomics Final Project

Riddhi Sera

2023-12-14

#1.Package Management and Setup

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(slingshot)
```

```
## Loading required package: princurve
## Loading required package: TrajectoryUtils
## Loading required package: SingleCellExperiment
## Warning: package 'SingleCellExperiment' was built under R version 4.3.2
## Loading required package: SummarizedExperiment
## Warning: package 'SummarizedExperiment' was built under R version 4.3.2
## Loading required package: MatrixGenerics
## Loading required package: matrixStats
##
## Attaching package: 'matrixStats'
## The following object is masked from 'package:dplyr':
##
##     count
##
## Attaching package: 'MatrixGenerics'
## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
```

```
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars

## Loading required package: GenomicRanges

## Loading required package: stats4

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min

## Loading required package: S4Vectors

## Warning: package 'S4Vectors' was built under R version 4.3.2

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:dplyr':
##
##     first, rename

## The following object is masked from 'package:utils':
##
##     findMatches

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Loading required package: IRanges

##
## Attaching package: 'IRanges'
```

```
## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice

## Loading required package: GenomeInfoDb

## Warning: package 'GenomeInfoDb' was built under R version 4.3.2

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

##
## Attaching package: 'Biobase'

## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians

## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
```

```
library(SingleCellExperiment)
library(Seurat)
```

```
## Loading required package: SeuratObject

## Loading required package: sp

##
## Attaching package: 'sp'

## The following object is masked from 'package:IRanges':
##
##     %over%

## 'SeuratObject' was built with package 'Matrix' 1.6.3 but the current
## version is 1.6.4; it is recomended that you reinstall 'SeuratObject' as
## the ABI for 'Matrix' may have changed

##
## Attaching package: 'SeuratObject'

## The following object is masked from 'package:SummarizedExperiment':
##
##     Assays

## The following object is masked from 'package:GenomicRanges':
##
##     intersect

## The following object is masked from 'package:GenomeInfoDb':
##
##     intersect

## The following object is masked from 'package:IRanges':
##
##     intersect
```

```
## The following object is masked from 'package:S4Vectors':
##
##     intersect
## The following object is masked from 'package:BiocGenerics':
##
##     intersect
## The following object is masked from 'package:base':
##
##     intersect
##
## Attaching package: 'Seurat'
## The following object is masked from 'package:SummarizedExperiment':
##
##     Assays
```

```
library(scater)
```

```
## Loading required package: scuttle
## Warning: package 'scuttle' was built under R version 4.3.2
## Loading required package: ggplot2
```

```
library(igraph)
```

```
##
## Attaching package: 'igraph'
## The following object is masked from 'package:scater':
##
##     normalize
## The following object is masked from 'package:Seurat':
##
##     components
## The following object is masked from 'package:GenomicRanges':
##
##     union
## The following object is masked from 'package:IRanges':
##
##     union
## The following object is masked from 'package:S4Vectors':
##
##     union
## The following objects are masked from 'package:BiocGenerics':
##
##     normalize, path, union
## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union
## The following objects are masked from 'package:stats':
##
```

```
##     decompose, spectrum
## The following object is masked from 'package:base':
##
##     union
library(leiden)
## create conda environment (yes/no)?
## no (use interactive mode)
## using environment: NA
## Unable to set up conda environment r-reticulate
## run in terminal:
## conda init
## conda create -n r-reticulate
## conda environment r-reticulate installed
## python modules igraph and leidenalg installed
library(Polychrome)
library(ggbeeswarm)
library(ggthemes)
```

#2.Data Preparation

```
#Reading the data
slingdata <- read.csv("Downloads/data.csv")
slingdata <- t(as.matrix(slingdata))

#Creating Seurat Object
slingseu <- CreateSeuratObject(counts = slingdata)
## Warning: Data is of class matrix. Coercing to dgCMatrix.
slingseu
## An object of class Seurat
## 16283 features across 7000 samples within 1 assay
## Active assay: RNA (16283 features, 0 variable features)
##  1 layer present: counts
#Data Normalization and Identification of Variable Features
all.genes <- rownames(slingseu)
#Normalizing
slingseu <- NormalizeData(slingseu, normalization.method = "LogNormalize")
## Normalizing layer: counts
#Finding Variable Features
slingseu <- FindVariableFeatures(slingseu, selection.method = "vst")
## Finding variable features for layer counts
top10 <- head(VariableFeatures(slingseu), 10)
#Scaling
slingseu <- ScaleData(slingseu, features = all.genes)
```

```
## Centering and scaling data matrix
#Printing top 10 variable genes
top10
```

```
##  [1] "SST"  "PPY"  "GCG"  "IAPP" "RBP4" "GHRL" "GAST" "PYY"  "INS"  "CCK"
```

#3.Principal Component Analysis (PCA)

```
slingseu <- RunPCA(slingseu, features = VariableFeatures(object = slingseu))
```

```
## PC_ 1
## Positive:  DCX, BASP1, MARCKS, MARCKSL1, RPL39, ONECUT2, CD24, MT.ND4L, MT.ND3, MAP1B
##     NREP, YWHAZ, SOX4, INSM1, FNBP1L, FARP1, DPYSL2, RAB3B, CADPS, TP53INP1
##     IGFBP5, APOC1, TMED8, BAALC, SOX11, FOXP1, RUNX1T1, PPP1CB, ACVR1C, GTF2I
## Negative:  B2M, SCG5, CD63, SCGB2A1, CHGB, MTRNR2L10, HLA.B, CYSTM1, HLA.A, CHCHD2
##     TIMP1, CD99, HLA.C, NAA20, CLU, RPL17, IGFBP7, QPCT, TUBA1B, TTR
##     CPE, FXYD2, MTRNR2L8, GNAS, NUPR2, RASD1, PEMT, SMIM22, FXYD5, ANXA2
## PC_ 2
## Positive:  RPS17, FTH1, PRSS23, MTRNR2L10, TUBA1A, HADH, RPL17, NPTX2, NEUROG3, CADM2
##     THSD4, OTULINL, RBP4, INS, CYYR1, C9orf16, DNAJC12, SAMD11, ZNF208, TUBB2B
##     PDX1, MT1F, MEX3A, TPPP3, TCTEX1D1, LRRIQ1, POU2F2, GSN, SHISAL2B, PKIG
## Negative:  GLS, TMEM238, DPP4, F10, SERPINI1, TTR, SPTSSB, BCAM, GC, SERPINA1
##     SLC7A2, IRX1, KIF12, ARRDC4, AGT, FOSB, ITPR1, C5orf38, CAMK2N1, GCG
##     RRBP1, ARFGEF3, SMIM32, CLU, ANK2, ELL2, IRX2, VSTM2A, TMEM236, JUND
## PC_ 3
## Positive:  ARX, GCG, KCTD12, IRX2, PRSS3, NEUROG3, TMSB4X, NR2F1, RGS4, C7
##     DEPP1, GC, TCTEX1D1, ZEB2, SPATS2L, SERPINA1, ALDH1A1, THSD4, ZNF208, EFCAB1
##     PDK3, GULP1, C9orf24, PRLHR, APOA1, C9orf16, C5orf38, MDK, KCND3, DUSP5
## Negative:  PCSK1, INS, DLK1, NKX6.1, CALB2, IAPP, SYT13, UCHL1, CHODL, EEF1A2
##     HADH, SLC17A6, NTM, PCDH7, ASCL2, PDX1, TUBB2B, LMO2, PARVB, KIF5C
##     GRIA4, ASB9, EBF1, NEFM, STMN4, SMAD9, LSAMP, CBLN1, PCSK1N, TMEM132A
## PC_ 4
## Positive:  COL5A2, TPH1, CBLN1, STC1, SLC18A1, ARPP21, FEV, LMX1A, CHST1, STAC
##     MGLL, AFF3, SORCS1, SYT6, BRINP3, MME, PRAG1, CHGA, ASTN2, MAP3K20
##     FAM162B, ZBTB7C, DDC, GPC4, KCNS3, GASK1B, STUM, NELL1, HDAC9, PDZRN4
## Negative:  ERO1B, INS, ISL1, PCDH7, ASPH, LMO2, ACVR1C, PLAGL1, IAPP, TMCC3
##     ST6GALNAC5, HPCA, DLK1, ASB9, TMOD1, PLPPR5, HADH, CHODL, ETV1, PDX1
##     CDH8, TENT5A, FAM13C, MAN1A1, CALB2, NEFM, PCP4, LHFPL6, SGCD, CASR
## PC_ 5
## Positive:  STMN4, FXYD6, TMSB10, CRYBA2, MDK, FABP5, FXYD5, TM4SF4, TMEM176B, TLCD3B
##     GNG3, GCG, C4orf48, ELAVL3, TMEM59L, MARCKSL1, NKAIN3, GPC2, CHRNA3, TMEM176A
##     BAALC, CD99, BTBD17, STMN1, ELAVL2, RPS17, LIMD2, LMO2, FXYD3, LOXL2
## Negative:  MTRNR2L12, SAMD11, TMEM238, PKIB, HADH, MT.CO2, MT.ATP6, CCND1, MTRNR2L8, MT.ND5
##     MT.CYB, CASR, G6PC2, PRSS23, ADCYAP1, GSN, ASB9, MT.ND3, ID1, FXYD2
##     MT.ND4L, PARVB, RBP4, DHRS2, ID3, FOSB, IAPP, INS, DAPL1, GAD2
```

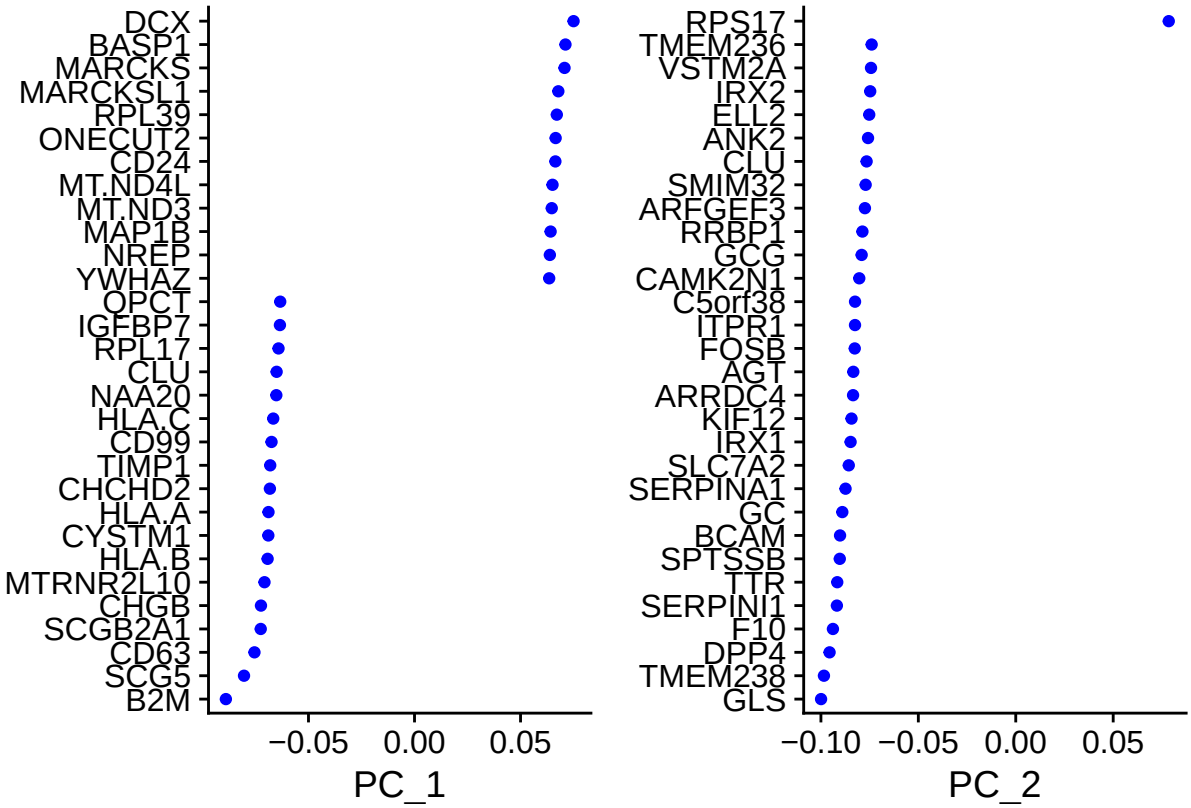
```
print(slingseu[["pca"]], dims = 1:5, nfeatures = 5)
```

```
## PC_ 1
## Positive:  DCX, BASP1, MARCKS, MARCKSL1, RPL39
## Negative:  B2M, SCG5, CD63, SCGB2A1, CHGB
## PC_ 2
## Positive:  RPS17, FTH1, PRSS23, MTRNR2L10, TUBA1A
## Negative:  GLS, TMEM238, DPP4, F10, SERPINI1
## PC_ 3
## Positive:  ARX, GCG, KCTD12, IRX2, PRSS3
```

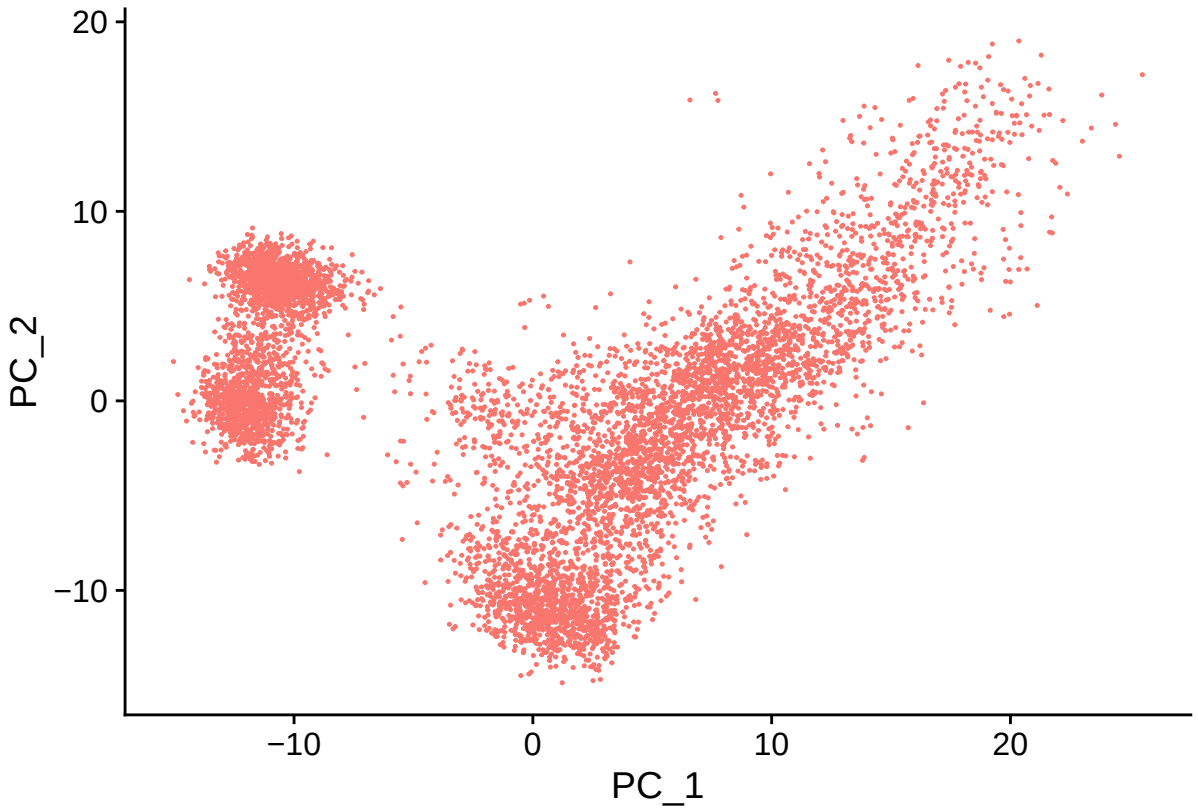
```
## Negative:  PCSK1, INS, DLK1, NKX6.1, CALB2
## PC_ 4
## Positive:  COL5A2, TPH1, CBLN1, STC1, SLC18A1
## Negative:  ERO1B, INS, ISL1, PCDH7, ASPH
## PC_ 5
## Positive:  STMN4, FXYD6, TMSB10, CRYBA2, MDK
## Negative:  MTRNR2L12, SAMD11, TMEM238, PKIB, HADH
```

#4.Visualising PCA Analysis

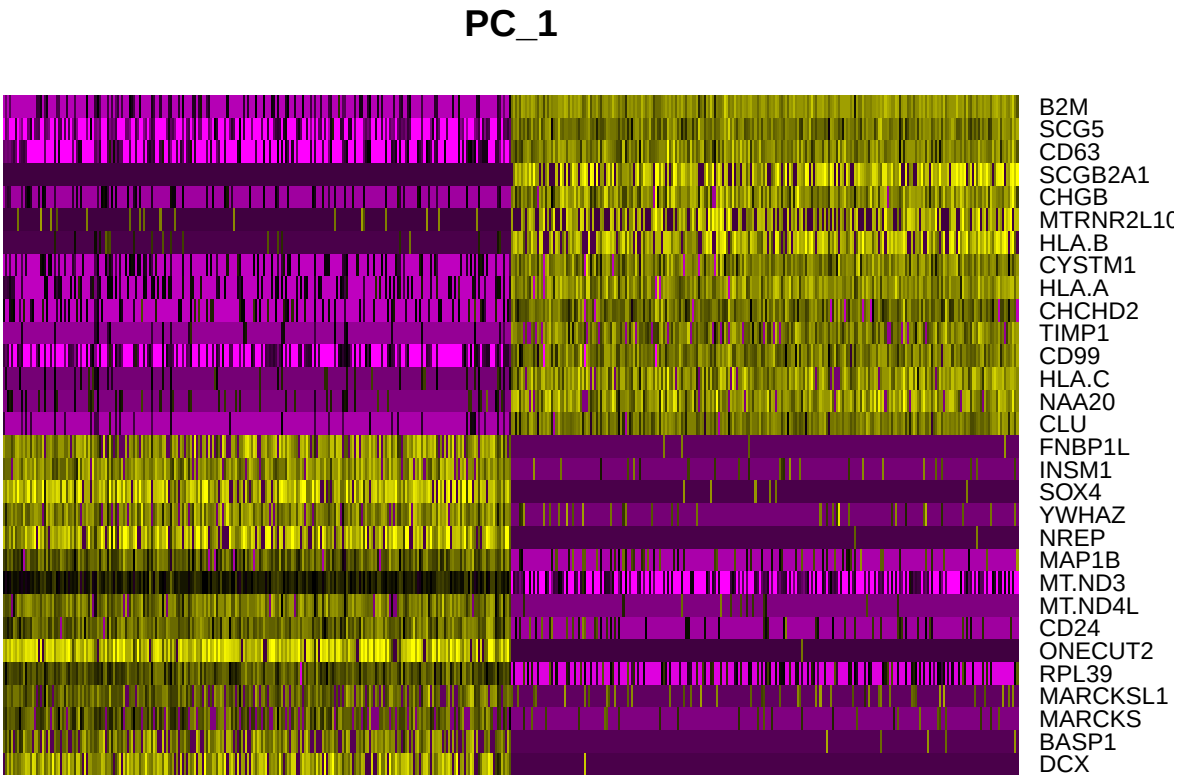
```
#Visualizes the contribution of each feature/gene for the first two principal
#components (PCs) of a PCA reduction
VizDimLoadings(slingseu, dims = 1:2, reduction = "pca")
```



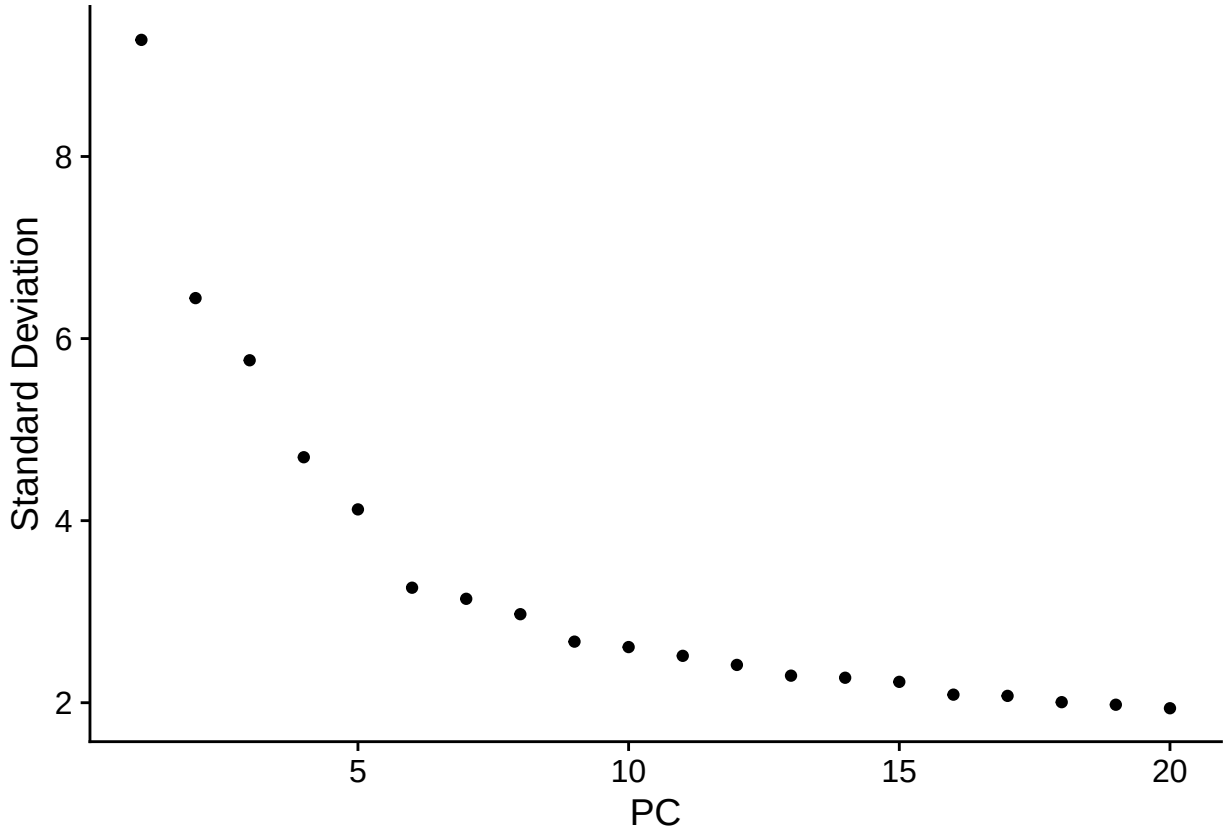

```
#Plots a 2D visualization of the PCA results
DimPlot(slingseu, reduction = "pca") + NoLegend()
```




```
#Generates a heatmap of the gene expression data focusing on the top genes
DimHeatmap(slingseu, dims = 1, cells = 500, balanced = TRUE)
```



```r
#Creates an elbow plot to determine the number of PCs to retain in the analysis
ElbowPlot(slingseu)
```




```r
#Computes a shared nearest neighbor (SNN) graph and identifies clusters
slingseu <- FindNeighbors(slingseu, dims = 1:10)
```

```
## Computing nearest neighbor graph
```

```
## Computing SNN
```

```r
slingseu <- FindClusters(slingseu, resolution = 0.5)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 7000
## Number of edges: 244577
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9363
## Number of communities: 15
## Elapsed time: 1 seconds
```

```r
head(Idents(slingseu), 5)
```

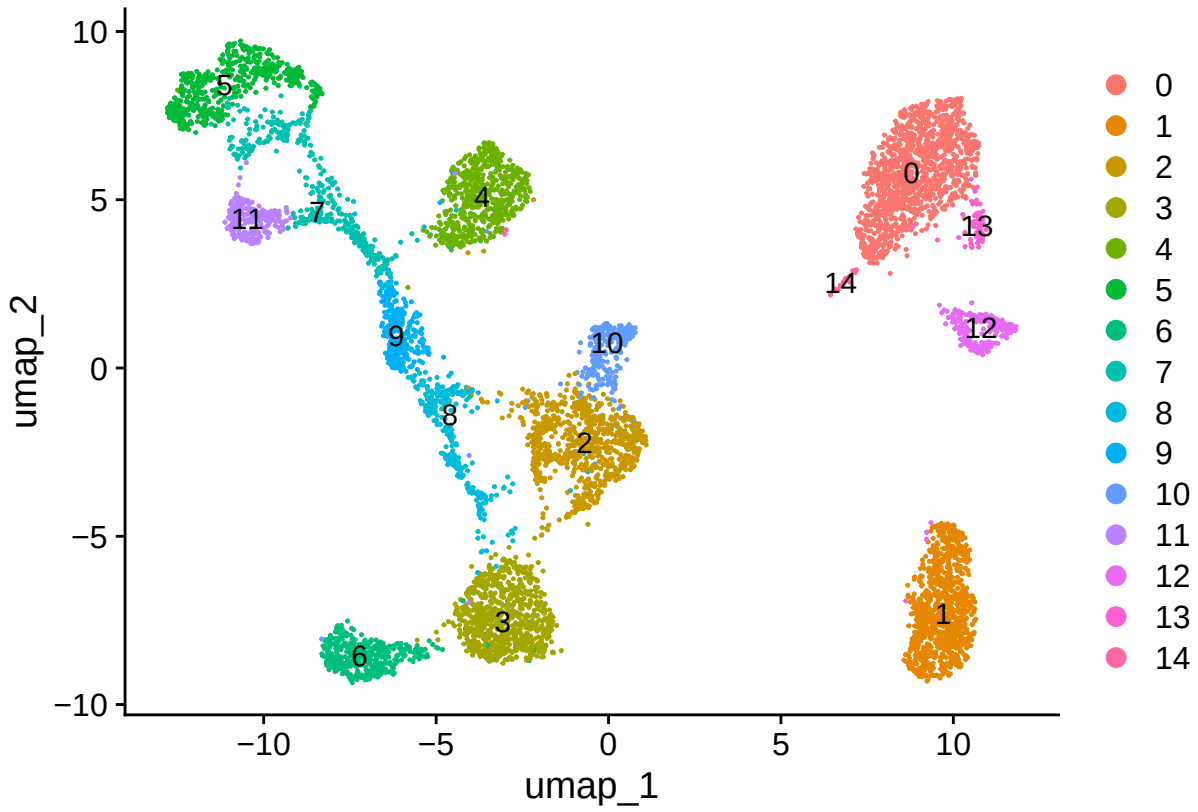
```
## Cell_1 Cell_2 Cell_3 Cell_4 Cell_5
##      0      1      1      4      5
## Levels: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
```

#5.UMAP

```r
slingseu <- RunUMAP(slingseu, dims = 1:10)
```

```
## Warning: The default method for RunUMAP has changed from calling Python UMAP via reticulate to the R
## To use Python UMAP via reticulate, set umap.method to 'umap-learn' and metric to 'correlation'
## This message will be shown once per session
## 02:38:16 UMAP embedding parameters a = 0.9922 b = 1.112
## Found more than one class "dist" in cache; using the first, from namespace 'BiocGenerics'
## Also defined by 'spam'
## 02:38:16 Read 7000 rows and found 10 numeric columns
## 02:38:16 Using Annoy for neighbor search, n_neighbors = 30
## Found more than one class "dist" in cache; using the first, from namespace 'BiocGenerics'
## Also defined by 'spam'
## 02:38:16 Building Annoy index with metric = cosine, n_trees = 50
## 0%   10   20   30   40   50   60   70   80   90   100%
## [----|----|----|----|----|----|----|----|----|----|
## **************************************************|
## 02:38:17 Writing NN index file to temp file /var/folders/92/y6ppswlj5dn165ls9hntjp1h0000gn/T//RtmpAA8
## 02:38:17 Searching Annoy index using 1 thread, search_k = 3000
## 02:38:19 Annoy recall = 100%
## 02:38:20 Commencing smooth kNN distance calibration using 1 thread with target n_neighbors = 30
## 02:38:21 Initializing from normalized Laplacian + noise (using RSpectra)
## 02:38:21 Commencing optimization for 500 epochs, with 288066 positive edges
## 02:38:31 Optimization finished
```

```r
DimPlot(slingseu, reduction = "umap", label = TRUE)
```

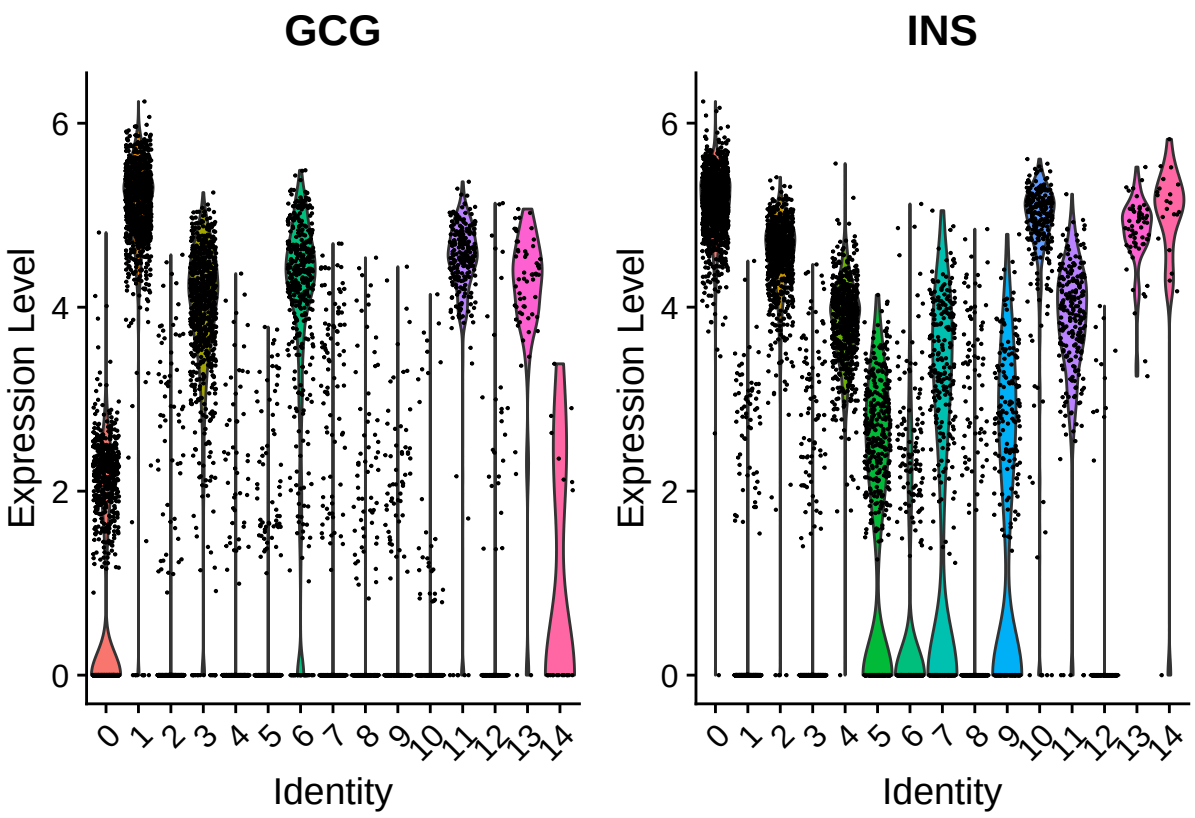


#6.Marker Gene Analysis

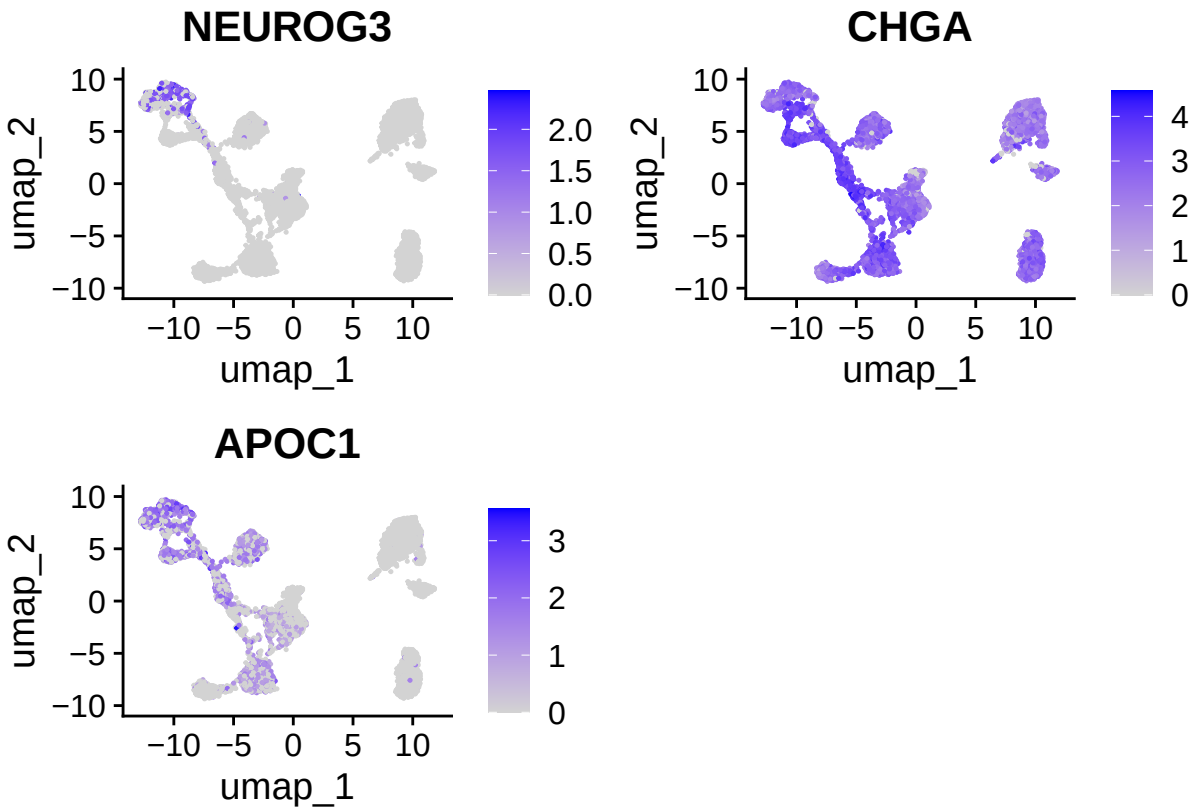
```
##identifying markers for each cluster
#cluster1.markers <- FindMarkers(slingseu, ident.1 = 2)
#head(cluster1.markers, n = 5)

##identifying markers for all clusters together
#slingseu.markers <- FindAllMarkers(slingseu, only.pos = TRUE)
```

#7.Gene Expression Visualisation

```
VlnPlot(slingseu, features = c("GCG", "INS"))
```




```
FeaturePlot(slingseu, features = c("NEUROG3", "CHGA", "APOC1"))
```



#8.Heatmap Visualization

```
#slingseu.markers %>%
#  group_by(cluster) %>%
#  dplyr::filter(avg_log2FC > 1) %>%
#  slice_head(n = 3) %>%
#  ungroup() -> top3
#DoHeatmap(slingseu, features = top3$gene) + NoLegend()
```

#9.Predicting Lineages with Slingshot

```
##We need to first convert it to a SingleCellExperiment object
sce <- as.SingleCellExperiment(slingseu)
reducedDims(sce)$UMAP <- Embeddings(slingseu, "umap")
colData(sce)$cluster <- Idents(slingseu)

#Perform slingshot analysis
sce <- slingshot(sce, clusterLabels = 'cluster', reducedDim = 'UMAP')

#Get lineage information
lnes <- getLineages(reducedDim(sce,"UMAP"), sce$ident)
print(lnes@metadata$lineages)
```

```
## $Lineage1
## [1] "0"  "13" "14" "10" "2"  "8"  "9"  "7"  "5"
##
## $Lineage2
## [1] "0"  "13" "14" "10" "2"  "8"  "9"  "7"  "11"
##
## $Lineage3
## [1] "0"  "13" "14" "10" "2"  "8"  "3"  "6"
##
## $Lineage4
## [1] "0"  "13" "14" "12" "1"
##
## $Lineage5
## [1] "0"  "13" "14" "4"
```

Now that we know that cluster 5 is most probably celltype S5, we shall choose cluster 5 as the initial cluster. This is in accordance with the last cluster number give by the getLineages function.

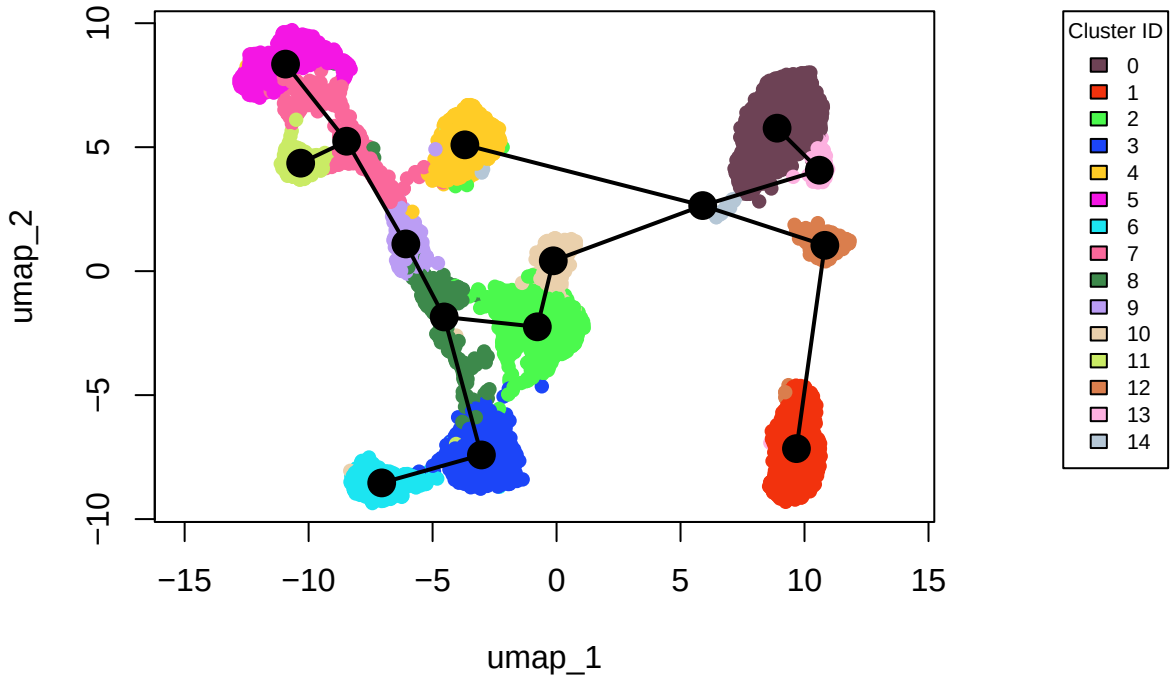
```
## seeding cluster 5 as the starting cluster
sce <- slingshot(sce, clusterLabels = 'cluster', reducedDim = "UMAP",
                 allow.breaks = FALSE, start.clus="5")

lnes <- getLineages(reducedDim(sce,"UMAP"), sce$ident, start.clus = "5")
print(lnes@metadata$lineages)
```

```
## $Lineage1
## [1] "5"  "7"  "9"  "8"  "2"  "10" "14" "13" "0"
##
## $Lineage2
## [1] "5"  "7"  "9"  "8"  "2"  "10" "14" "12" "1"
##
## $Lineage3
## [1] "5"  "7"  "9"  "8"  "2"  "10" "14" "4"
##
## $Lineage4
```

```
## [1] "5" "7" "9" "8" "3" "6"
##
## $Lineage5
## [1] "5"  "7"  "11"
```

#10. Visualizing the pseudotime or lineages

```
#Defining the cluster colors
# Assuming 'sce' is a SingleCellExperiment object with UMAP coordinates and cluster IDs
umap_coords <- reducedDims(sce)$UMAP
cluster_ids <- sce$ident

# Define the cluster colors
my_color <- createPalette(length(levels(cluster_ids)),
                          c("#010101", "#ff0000"), M=1000)
names(my_color) <- levels(cluster_ids)

par(mar=c(5.1, 4.1, 4.1, 8.1))
plot(umap_coords, col = my_color[as.character(cluster_ids)], pch=16, asp = 1)

centroids <- aggregate(reducedDims(sce)$UMAP,
                       by=list(cluster=as.factor(sce$ident)), FUN=mean)
text(centroids[,2], centroids[,3], labels=names(centroids$cluster), cex=0.8,
     pos=3, col="white", bg="black")

lines(SlingshotDataSet(lnes), lwd=2, type='lineages', col="black")

legend("topright", inset=c(-0.3,0), legend = names(my_color),
       fill = my_color, cex=0.7, xpd=TRUE, horiz=FALSE, title="Cluster ID")
```




```
col_data <- colData(sce)
num_rows <- nrow(col_data)
num_cols <- length(col_data)
```

```r
slingshot_df <- data.frame(matrix(nrow = num_rows, ncol = num_cols))
for (i in seq_along(col_data)) {
  slingshot_df[[i]] <- col_data[[i]]
}

colnames(slingshot_df) <- names(col_data)
slingshot_df$ident = factor(slingshot_df$ident, levels=c(5,7,9,8,2,10,14,13,0))
names(slingshot_df) <- make.unique(names(slingshot_df))
ggplot(slingshot_df, aes(x = names(slingshot_df)[14], y = as.factor(ident),
                         colour = as.factor(ident))) +
  geom_quasirandom(groupOnX = TRUE, alpha = 0.7, size = 1.5) +
  theme_classic() +
  theme(
    legend.position = "right",
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.title = element_text(size = 12),
    legend.title = element_text(size = 10),
    legend.text = element_text(size = 8)
  ) +
  scale_color_brewer(palette = "Set3") +
  xlab("First Slingshot pseudotime") +
  ylab("Cell Type") +
  ggtitle("Cells ordered by Slingshot Pseudotime") +
  labs(colour = "Identity")
```

```
## Warning: Removed 3185 rows containing missing values (`geom_point()`).
```